Cloud computing versus grid computing

Service types, similarities and differences, and things to consider

Judith M. Myerson

March 03, 2009

Want to know more about cloud and grid computing? Learn how you can use Infrastructure as a Service to get a full computer infrastructure using Amazon's Elastic Compute Cloud (EC2). See the similarities, differences, and issues to consider in grid and cloud computing. Explore some of the security issues and choices for Web development in the cloud, and see how you can be environmentally friendly using cloud computing.

Introduction

You may have wondered about cloud computing as compared to grid computing. In this article, I talk about cloud computing service types and the similarities and differences between cloud and grid computing. I look at why cloud computing may be advantageous over grid computing, what issues to consider in both, and some security concerns. I use Amazon Web Services as an example.

To get cloud computing to work, you need three things: thin clients (or clients with a thick-thin switch), grid computing, and utility computing. Grid computing links disparate computers to form one large infrastructure, harnessing unused resources. Utility computing is paying for what you use on shared servers like you pay for a public utility (such as electricity, gas, and so on).

With grid computing, you can provision computing resources as a utility that can be turned on or off. Cloud computing goes one step further with on-demand resource provisioning. This eliminates over-provisioning when used with utility pricing. It also removes the need to over-provision in order to meet the demands of millions of users.

Infrastructure as a Service and more

A consumer can get service from a full computer infrastructure through the Internet. This type of service is called Infrastructure as a Service (IaaS). Internet-based services such as storage and databases are part of the IaaS. Other types of services on the Internet are Platform as a Service (PaaS) and Software as a Service (SaaS). PaaS offers full or partial application development that users can access, while SaaS provides a complete turnkey application, such as Enterprise Resource Management through the Internet.
To get an idea of how Infrastructure as a Service (IaaS) is used in real life, consider *The New York Times* that processed terabytes of archival data using hundreds of Amazon's EC2 instances within 36 hours. If *The New York Times* had not used EC2, it would have taken it days or months to process the data.

The IaaS divides into two types of usage: public and private. Amazon EC2 uses public server pools in the infrastructure cloud. A more private cloud service uses groups of public or private server pools from an internal corporate data center. You can use both types to develop software within the environment of the corporate data center, and, with EC2, temporarily extend resources at low cost—say for testing purposes. The mix may provide a faster way of developing applications and services with shorter development and testing cycles.

### Amazon Web services

With EC2, customers create their own Amazon Machine Images (AMIs) containing an operating system, applications, and data, and they control how many instances of each AMI run at any given time. Customers pay for the instance-hours (and bandwidth) they use, adding computing resources at peak times and removing them when they are no longer needed. The EC2, Simple Storage Service (S3), and other Amazon offerings scale up to deliver services over the Internet in massive capacities to millions of users. Amazon provides five different types of servers ranging from simple-core x86 servers to eight-core x86_64 servers. You do not have to know which servers are in use to deliver service instances. You can place the instances in different geographical locations or availability zones. Amazon allows elastic IP addresses that can be dynamically allocated to instances.

### Cloud computing

With cloud computing, companies can scale up to massive capacities in an instant without having to invest in new infrastructure, train new personnel, or license new software. Cloud computing is of particular benefit to small and medium-sized businesses who wish to completely outsource their data-center infrastructure, or large companies who wish to get peak load capacity without incurring the higher cost of building larger data centers internally. In both instances, service consumers use what they need on the Internet and pay only for what they use.

The service consumer no longer has to be at a PC, use an application from the PC, or purchase a specific version that's configured for smartphones, PDAs, and other devices. The consumer does not own the infrastructure, software, or platform in the cloud. He has lower upfront costs, capital expenses, and operating expenses. He does not care about how servers and networks are maintained in the cloud. The consumer can access multiple servers anywhere on the globe without knowing which ones and where they are located.

### Grid computing

Cloud computing evolves from grid computing and provides on-demand resource provisioning. Grid computing may or may not be in the cloud depending on what type of users are using it. If the users are systems administrators and integrators, they care how things are maintained in the...
cloud. They upgrade, install, and virtualize servers and applications. If the users are consumers, they do not care how things are run in the system.

Grid computing requires the use of software that can divide and farm out pieces of a program as one large system image to several thousand computers. One concern about grid is that if one piece of the software on a node fails, other pieces of the software on other nodes may fail. This is alleviated if that component has a failover component on another node, but problems can still arise if components rely on other pieces of software to accomplish one or more grid computing tasks. Large system images and associated hardware to operate and maintain them can contribute to large capital and operating expenses.

**Similarities and differences**

Cloud computing and grid computing are scalable. Scalability is accomplished through load balancing of application instances running separately on a variety of operating systems and connected through Web services. CPU and network bandwidth is allocated and de-allocated on demand. The system’s storage capacity goes up and down depending on the number of users, instances, and the amount of data transferred at a given time.

Both computing types involve multitenancy and multitask, meaning that many customers can perform different tasks, accessing a single or multiple application instances. Sharing resources among a large pool of users assists in reducing infrastructure costs and peak load capacity. Cloud and grid computing provide service-level agreements (SLAs) for guaranteed uptime availability of, say, 99 percent. If the service slides below the level of the guaranteed uptime service, the consumer will get service credit for receiving data late.

The Amazon S3 provides a Web services interface for the storage and retrieval of data in the cloud. Setting a maximum limits the number of objects you can store in S3. You can store an object as small as 1 byte and as large as 5 GB or even several terabytes. S3 uses the concept of buckets as containers for each storage location of your objects. The data is stored securely using the same data storage infrastructure that Amazon uses for its e-commerce Web sites.

While the storage computing in the grid is well suited for data-intensive storage, it is not economically suited for storing objects as small as 1 byte. In a data grid, the amounts of distributed data must be large for maximum benefit.

A computational grid focuses on computationally intensive operations. Amazon Web Services in cloud computing offers two types of instances: standard and high-CPU.

**Issues to consider**

Four issues stand out with cloud and grid computing: threshold policy, interoperability issues, hidden costs, and unexpected behavior.

**Threshold policy**

Let’s suppose I had a program that did credit card validation in the cloud, and we hit the crunch for the December buying season. Higher demand would be detected and more instances would be
created to fill that demand. As we moved out of the buying crunch, the need would be diminished and the instances of that resource would be de-allocated and put to other use.

To test if the program works, develop, or improve and implement, a threshold policy in a pilot study before moving the program to the production environment. Check how the policy detects sudden increases in the demand and results in the creation of additional instances to fill in the demand. Also check to determine how unused resources are to be de-allocated and turned over to other work.

Interoperability issues

If a company outsources or creates applications with one cloud computing vendor, the company may find it is difficult to change to another computing vendor that has proprietary APIs and different formats for importing and exporting data. This creates problems of achieving interoperability of applications between these two cloud computing vendors. You may need to reformat data or change the logic in applications. Although industry cloud-computing standards do not exist for APIs or data import and export, IBM and Amazon Web Services have worked together to make interoperability happen.

Hidden costs

Cloud computing does not tell you what hidden costs are. For instance, companies could incur higher network charges from their service providers for storage and database applications containing terabytes of data in the cloud. This outweighs costs they could save on new infrastructure, training new personnel, or licensing new software. In another instance of incurring network costs, companies who are far from the location of cloud providers could experience latency, particularly when there is heavy traffic.

Unexpected behavior

Let's suppose your credit card validation application works well at your company's internal data center. It is important to test the application in the cloud with a pilot study to check for unexpected behavior. Examples of tests include how the application validates credit cards, and how, in the scenario of the December buying crunch, it allocates resources and releases unused resources, turning them over to other work. If the tests show unexpected results of credit card validation or releasing unused resources, you will need to fix the problem before running the application in the cloud.

Security issues

In February 2008, Amazon's S3 and EC2 suffered a three-hour outage. Even though an SLA provides data recovery and service credits for this type of outage, consumers missed sales opportunities and executives were cut off from critical business information they needed during the outage.

Instead of waiting for an outage to occur, consumers should do security testing on their own—checking how well a vendor can recover data. The test is very simple. No tools are needed. All you have to do is to ask for old data you have stored and check how long it takes for the vendor to
recover. If it takes too long to recover, ask the vendor why and how much service credit you would get in different scenarios. Verify if the checksums match the original data.

An area of security testing you should do is to test a trusted algorithm to encrypt the data on your local computer, and then try to access data on a remote server in the cloud using the decryption keys. If you can't read the data once you have accessed it, the decryption keys are corrupted, or the vendor is using its own encryption algorithm. You may need to address the algorithm with the vendor.

Another issue is the potential for problems with data in the cloud. To protect the data, you may want to manage your own private keys. Check with the vendor on the private key management. Amazon will give you the certificate if you sign up for it.

Software development in cloud

To develop software using high-end databases, the most likely choice is to use cloud server pools at the internal data corporate center and extend resources temporarily with Amazon Web services for testing purposes. This allows project managers to better control costs, manage security, and allocate resources to clouds a project is assigned to. The project managers could also assign individual hardware resources to different cloud types: Web development cloud, testing cloud, and production cloud. The cost associated with each cloud type may differ from one another. The cost per hour or usage with the development cloud is most likely lower than the production cloud, as additional features, such as SLA and security, are allocated to the production cloud.

The managers can limit projects to certain clouds. For instance, services from portions of the production cloud can be used for the production configuration. Services from the development cloud can be used for development purpose only. To optimize assets at varying stages of the project of software development, the managers can get cost-accounting data by tracking usage by project and user. If the costs are found to be high, managers can use Amazon EC2 to temporarily extend resources at a very low cost provided that security and data recovery issues have been resolved.

Environmentally friendly cloud computing

One incentive for cloud computing is that it may be more environmentally friendly. First, reducing the number of hardware components needed to run applications on the company's internal data center and replacing them with cloud computing systems reduces energy for running and cooling hardware. By consolidating these systems in remote centers, they can be handled more efficiently as a group.

Second, techniques for cloud computing promote telecommuting techniques, such as remote printing and file transfers, potentially reducing the need for office space, buying new furniture, disposing of old furniture, having your office cleaned with chemicals and trash disposed, and so on. They also reduce the need for driving to work and the resulting carbon dioxide emissions.
Conclusion

This article helps you plan ahead for working with cloud by knowing how cloud computing compares to grid computing, how you can resolve issues in cloud and grid computing, and what security issues exist with data recovery and managing private keys in a pay-on-demand environment. Potential consumers' demands for increased capacities over the Internet present a challenge for the developers and other members of a project team. Being aware of and resolving the issues of Web application design and potential security issues can make your team's experiences trouble-free. To help, look at IBM Rational Web Developer WebSphere Software to build Web applications and IBM Rational ClearQuest for defect and application tracking (see Related topics).
Related topics

- Learn more about IBM's Cloud Services Initiatives.
- Find out more about IBM's Grid Computing.
- The Work with Web services in enterprise-wide SOA series by Judith M. Myerson offers information on how to work with Web services in enterprise-wide SOAs.
- Browse the Judith M. Myerson's series, Use SLAs in a Web services context, and get details on service-level agreements.
- Want more information on Ajax tools? Read about them in "Survey of Ajax tools and techniques" (Gal Shachor, Yoav Rubin, Shmulik London, Shmuel Kallner, developerWorks, July 2007).
- Read "Tight coupling Web services in the SOA" (developerWorks, Jan 2008).
- Read Judith M. Myerson's The Complete Book of Middleware, which focuses on the essential principles and priorities of system design and emphasizes the new requirements brought forward by the rise of e-commerce and distributed integrated systems.
- Get the business insight and the technical know-how to ensure successful systems integration by reading Enterprise Systems Integration, Second Edition.
- Bring your organization into the future with RFID in the Supply Chain, which explains business processes, operational and implementation problems, risks, vulnerabilities, and security and privacy.
- IBM Redbooks: Read Tivoli Manager for Domino V2.1 Fulfilling Service-Level Agreements Using Tivoli Technology, for IBM Lotus Domino administrators, which goes into the nuts and bolts of developing a service-level agreement.
- The IBM SOA Web site offers an overview of SOA and how IBM can help you get there.
- See how IBM Rational Web Developer for WebSphere Software for architecture management, IBM Rational ClearQuest for change and release management, and IBM Rational Functional Tester Plus for quality management can help when developing Ajax and other applications. These tools from IBM help increase your productivity by reducing testing time and the costs of test labs in your enterprise.
- IBM trial products for download: Build your next development project with IBM trial software, available for download directly from developerWorks.